Strategies for Security Measurement Objective Decomposition

Reijo M. Savola
VTT Technical Research Centre of Finland
Oulu, Finland
reijo.savola@vtt.fi

***Abstract*—Systematically managed, sufficient and credible security metrics increase the understanding of the security effectiveness level of software-intensive systems during the system development and operation. Risk-driven top-down modeling enables systematic and meaningful security metrics development. We propose six strategies for security measurement objective decomposition. Their focus is on metrics development for security correctness, software and system quality, partial security effectiveness, as well as security-related compliance and tradeoff decision-making. The proposed strategies integrate an abstract security effectiveness model, security measurement objectives, and the associated measurement points in relevant system components. Security effectiveness is emphasized in all strategies despite of other objectives.**

Keywords-security metrics; decomposition; security effectiveness; security correctness; system quality

I. Introduction

As software-intensive systems incorporate increasingly critical applications, are more difficult to manage, and utilize more complex and networked software, they become exposed to an increasing number of security risks. There is a need for practical and systematic techniques with which to obtain sufficient and credible evidence of the operational systems' SE (Security Effectiveness). There are already a variety of security metrics proposed in the literature, as summarized e.g. in [1][2][3][4]. However, one of the most important questions still remains unsolved: how to manage the metrics in a way that the can offer meaningful input to security decision-making. Quantification and decomposition techniques are widely used in engineering as means to increase understanding of complex systems, to plan activities in better manageable components, and to enable informed decision-making. Research of the utilization of these techniques in security is still in its infancy due to the complex and wide nature of security phenomena. Security engineering requires expertise from several domains like systems engineering, software engineering, risk management, business management and information security management. A horizontal security metrology model along with usable abstractions to control security-related information flow is needed to integrate the essential knowledge from different expert areas for the purposes of secure systems engineering. SMOs (Security Measurement Objectives) are the essential objectives for security measurement. Typically they emphasize SE [5][6][7]. Because of the variety of system architectures, it is not practical to define a general-purpose SMO decomposition model. However, an SMO decomposition model tailored to the system architecture helps to relate low-level metrics and measurements to high-level goals and requirements.

The main contribution of this study is in proposing strategies for SMO decomposition. A decomposition strategy is a heuristics method for selecting subsets of attributes to be assigned to sub-objectives. The strategies are chosen based on typical security engineering and management needs from software-intensive system development and operation. Section II briefly discusses the background of this study. Section III proposes a generic methodology, Section IV applies it for basic strategies for configuration correctness, partial direct SE, and security-relevant software and system quality, and Section V to integrated strategies with several measurement goals. Section VI discusses related work and Section VII offers concluding remarks and discusses future research questions.

II. Background

Table I summarizes the generic terminology used in this study. Sometimes security correctness is referred to as *security accuracy*, since total correctness is very challenging to achieve.

TABLE I. Summary of Generic Terminology in this Study

Term	**Rf.**	**Explanation**
Security control	[5]	Means of managing risk, which can be administrative, technical, management, or legal in nature.
Security effective-ness	[6] [7] [8]	Assurance that the stated security objectives are met in the SuI and the expectations for resiliency in the use environment are satisfied, while the SuI does not behave other than intended.
Security correct-ness	[6] [7] [8]	Assurance that the security controls have been correctly deployed in the SuI. In practice, the aim is to measure how close the actual deployed security controls in a real system are to their ideal deployment as defined by the security objectives.
Security efficiency	[6]	Assurance that the adequate security quality has been achieved in the SuI, meeting the resource, time and cost constraints.
Security objective	[9]	High-level statements of intent to counter identified threats and/or satisfy identified organizational security policies and/or assumptions.

We use the following SMO classification in this study: SEMO (Security Effectiveness Measurement Objective), SEyMO (Security Efficiency Measurement Objective), RMO (Regulatory Requirement Measurement Objective), and BPMO (Best Practice Measurement Objective). RMOs and BPMOs are categories of SCMOs (Security Correctness Measurement Objectives).

A. Factors contributing to SE

From technical perspective, SE can be measured with large accuracy only during long periods of the actual operation of the system, when it is exposed to real *security risk occurrence*. Penetration testing is often used to obtain evidence of SE during the late phases of system development. For the above reasons, direct SE measurement can only be *partial*. Direct partial SE, security correctness, and software and system quality are different *security measurement spaces*, yet dependent on each other. All of them should contribute to the perceived level of SE. It is not an exact measurable property, yet factors contributing to it can be measured, see Fig. 1.

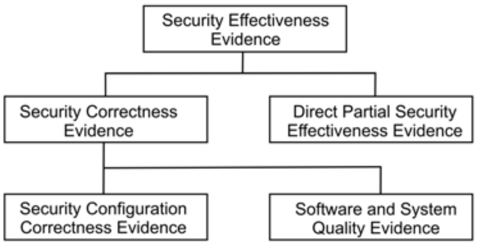
Security Effectiveness Evidence
Security Correctness Evidence
Direct Partial Security Effectiveness Evidence
Security Configuration Correctness Evidence
Software and System Quality Evidence

Figure 1. Example security evidence contributing to SE.

Security correctness is a key factor that contributes to SE of the SuI (System under Investigation) due to its concreteness. However, it must be noted that it does not automatically imply SE, and proper RA (Risk Analysis) or at least using best practices is required. The quality of RA has a crucial role in the definition and maintenance of security objectives.

Unfortunately, security measurement based on security controls is not enough for practical systems. Hayden [10] claims that the security control concept is myopic, and McGraw [11] adds that a software security problem is more likely to arise because of a problem in a *system's standard-issue part* than in a mechanism implemented based on security objectives. Moreover, design specifications often miss important security details that appear only in code [12]. Because of these challenges, security-relevant system quality (correctness) should be investigated in practical systems. Software quality is more in focus compared to hardware quality because software behavior and performance characteristics are different than those experienced from a dependability perspective: whereas most hardware component failure data are well documented and experienced in use environment, the nature of software faults and their traceability of cause and effects are not easy to determine [13]. Therefore, we emphasize software quality, and denote this measurement space as *software and system quality*. Adequate SW&SQ (Software and System Quality) of the SuI s a central objective in minimizing the security vulnerabilities arising from the SuI, *which cannot be managed* by the security controls only.

To distinguish between the system as a whole (SuI) and its security controls, we use the term CuI (Control under Investigation). Security correctness of a CuI depends on the security configuration correctness and the adequate SW&SQ associated to the CuI.

B. Hierarchical Security Metrics Development

A hierarchical security metrics development methodology utilizing security requirement decomposition was introduced in [14] and [15]. The work discussed security threats and security requirements for the GEMOM (Genetic Message Oriented Middleware) [16] system. Furthermore, it introduced a collection of security metrics development heuristics for correctness and effectiveness of selected security controls. The heuristics were expressed in a form of BMCs (Basic Measurable Components), leaf components of the system's security requirement decomposition that clearly manifest a measurable property of the system. Term BM (Base Measure) is often used for BMCs.

III. PROPOSED GENERIC METHODOLOGY

In the following, we propose a generic methodology SMO for decomposition strategies. The detailed strategies of Section IV are based on this methodology.

An SMO consists of a top-level SMO and its constituent sub-objectives, and at the leaf level, BMCs, for which, detailed security metrics can be developed. The goal of high-quality decomposition process is to maintain traceability and sufficient equivalency of the original objectives, SMOs, sub-objectives and metrics resulting from the decomposition. In practice, there are gaps and biases; sub-objectives contain modified versions of the original objectives. Although decompositions are often based on ad-hoc analysis, they systematize and support management security decision-making.

A. Security Effectiveness Abstract Model

A core component of our generic methodology is SEAM (Security Effectiveness Abstract Model), a simplified model that contains the core knowledge of factors *contributing* to the SE of the CuI. An SEAM is used to *guide* SMO decomposition effort. Fig. 2 proposes a basic structure of SEAM.

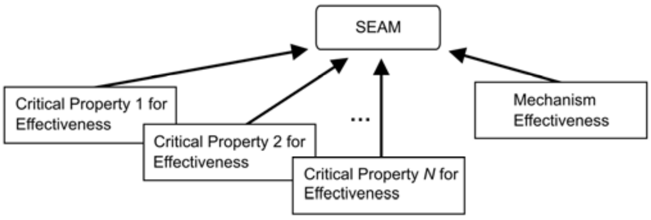
SEAM
Critical Property 1 for Effectiveness
Critical Property 2 for Effectiveness
...
Critical Property N for Effectiveness
Mechanism Effectiveness

Figure 2. Basic structrure of SEAM.

The model should be abstract enough to be applicable to all dimensions: it does not directly address the system's architectural components, correctness, quality or partial effectiveness. These issues are included in the other parts of the methodology. Note that the type of core knowledge varies based on the security objectives, security requirements and compliance requirements, and may be updated based on RA and measurement results. The SEAM will be used as a reference model in more detailed metrics development, taking into account the system architecture. In Fig. 2, SE of the mechanism (e.g., an CuI) is shown as its own decomposition branch, as important as other branches. The other branches are added to emphasize critical properties for SE.

Fig. 3 shows example (high-level) SEAMs for authentication, authorization, confidentiality, integrity, availability, and non-repudiation. Critical properties are illustrated by parallelograms, and *nested hierarchy of other SEAMs* by boxes with dashed lines. For example, the reasoning behind the Authentication SEAM in the figure is that both the authentication *mechanism* and the *identity strength* contribute essentially to authentication effectiveness. Authentication effectiveness is linked to access control effectiveness. This linkage supports the provision of fine-grained access control in the SuI where different applications have varied authentication requirements. A stronger authentication should grant a user with higher privileges, within the range of authorized ones.

B. Trust Assumptions, Trust Values for Unmeasured Parts

It is not always possible to obtain measured evidence of the CuI; the CuI becomes *unmeasured*. For example, the CuI or part of it can be *unmanaged* [17] from the system administration perspective: that part is not in the control of the party who is utilizing security measurements. In the unmeasured parts, in the absence of measured evidence, there may be a certain amount of trust that the security level of the object is at an adequate level. It is possible to make *trust assumption* that that part contributes to SE at an adequate level, taking into account its interdependencies with other parts of the

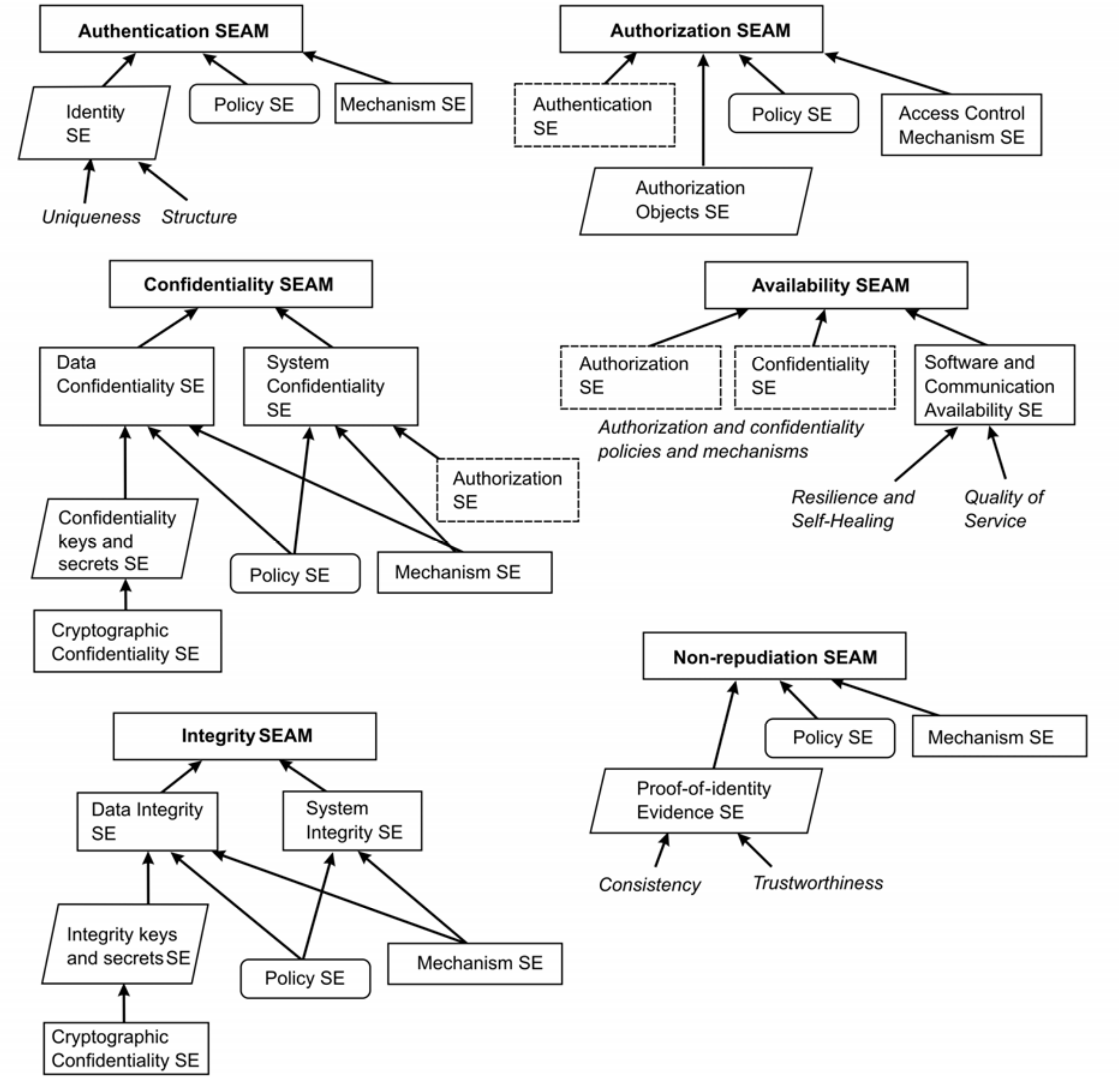

Figure 3. Examples of high-level SEAMs for selected security controls.

SuI. This trust assumption is based on, e.g., security assurance claims carried out by a representative of the unmanaged object or a third party, or on subjective reputation parameters. The actual values resulting from trust assumptions are *trust values*.

C. Generic Methodology

The decomposition strategies proposed in the following section are structured according to the generic methodology displayed in Fig. 4.

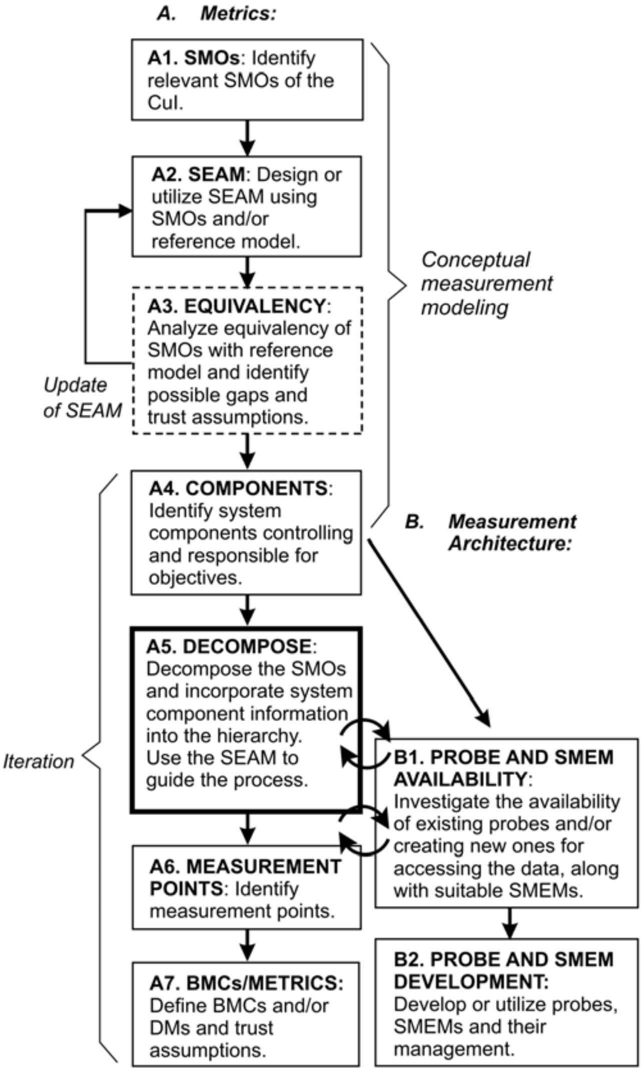
A. Metrics:
A1. SMOs: Identify relevant SMOs of the CuI.
A2. SEAM: Design or utilize SEAM using SMOs and/or reference model.
A3. EQUIVALENCY: Analyze equivalency of SMOs with reference model and identify possible gaps and trust assumptions.
Update of SEAM
Conceptual measurement modeling
A4. COMPONENTS: Identify system components controlling and responsible for objectives.
B. Measurement Architecture:
A5. DECOMPOSE: Decompose the SMOs and incorporate system component information into the hierarchy. Use the SEAM to guide the process.
Iteration
B1. PROBE AND SMEM AVAILABILITY: Investigate the availability of existing probes and/or creating new ones for accessing the data, along with suitable SMEMs.
A6. MEASUREMENT POINTS: Identify measurement points.
A7. BMCs/METRICS: Define BMCs and/or DMs and trust assumptions.
B2. PROBE AND SMEM DEVELOPMENT: Develop or utilize probes, SMEMs and their management.

Figure 4. Generic methodology for SMO decomposition strategies.

The metrics development stages are shown in Branch A, and measurement architecture stages in Branch B. The methodology starts from the identification of core SMOs of the CuI. SMOs, especially SEOs, should be based on adequate prioritized RA results. In Stage A2, the SEAM will be developed, or alternatively, the suitability of a pre-existing SEAM is analyzed and possible modifications are done to it. The optional Stage A3 (Equivalency) is needed when another core reference model (e.g. standard or regulation) involved. During this step, possible priority conflicts between SMOs, SEAM and the reference model objectives are analyzed and solved. The results are updated to the SEAM. The actual decomposition effort is in Stages A4–A7, aiming at BMCs and DMs. The process is iterated, incorporating SMOs and components associated to them. The SEAM is used as heuristics to guide the actual decomposition task. Measurement points (Stage A6) are points for actual measurement data gathering from the system, e.g. probe inputs or data fields. Implementation and deployment of SMEMs (Security Measurability Enhancing Mechanisms), mentioned in Stages B1 and B2, [19] increases the feasibility of security measurements, and availability and attainability of evidence.

The root node of the decomposition is the SuI itself, with the main SMOs placed in nodes below it, in priority order. The main phases of the decomposition process are [14]:

1. Identify successive components from each goal that *contribute essentially* to the SMO.
2. Examine the subordinate nodes to determine whether further decomposition is needed. If it is, repeat the process with the subordinate nodes as current goals, breaking them down to their essential components, and
3. Terminate the decomposition process when none of the leaf nodes can be decomposed any further, or when further analysis of these components is no longer necessary.

IV. Proposed Basic SMO Decomposition Strategies

A. (Security) Configuration Correctness

Table II shows the proposed decomposition strategy for this category, structured according to the stages of Fig. 4.

TABLE II. STRATEGY 1: CONFIGURATION CORRECTNESS OF CUI

Stage	Additions to the general methodology
A1	SMOs are SCMOs. Even though the goal is correctness, define them emphasizing SE as much as possible. Investigate available and attainable security configuration evidence, and their relevance to SE. Prioritize the results with respect to SE.
A2–A3	Develop an SEAM, which incorporates heuristics for security configuration correctness and/or system correctness, including the correct deployment control of the CuI.
A4	Identify system components relevant to SCMOs in the metrics hierarchy being developed. The components are architectural components (like modules, devices, protocols, interfaces, platforms) where the CuI correctness is is configured or deployment of the CuI is enabled.
A6	Identify measurement points in the metrics hierarchy. They are data structures, devices or files where the configuration data and deployment control resides.
A7	In the decomposition, BMCs should aim at feasible metrics or use of available metrics, aiming at ‘ok’ or ‘not ok’ conclusion. Detailed reference requirements can be used to define the correctness criteria.

Most SCMOs belong to this category. Correct configuration, including correct CuI deployment, should be investigated regardless of what it the origin of correctness requirement (regulation, risk management decision or best practice). This SMO category is *more concrete* than many others, and is an goal for many other types of security metrics. In its simplest form, the correctness measurement result is either ‘ok’ or ‘not ok’ – in these metrics, the configuration can be either adequate or not adequate. The aim of the strategy is to enable development of metrics for security CC (Configuration Correctness) and/or system CC.

Example (*High-level plan for CC with identified system components*) Fig. 5 shows an example high-level plan for SCMO decomposition in GEMOM [16] end-user authentication. The figure illustrates the relevant authentication mechanisms and associated main components. The plan is based on the Authentication SEAM of Fig. 3, focusing on identity strength. The end-user authentication in GEMOM system is based on managed *iCards*. An iCard is always associated to an identity, managed by the IdP (Identity Provider) providing the card. The authentication transaction between the authentication client and the IdP is based on strong cryptographic protocols, utilizing the WS-* (Web Services) [20] family of standards. An iCard in GEMOM can be based on (*i*) a password, (*ii*) a software stored X.509 certificate [21], or (*iii*) an X.509 certificate on a smart card, creating three different levels of authentication strength. In (*ii*) and (*iii*), the user's X.509 certificate replaces the user-specific secret [22]. The actual measurement points are omitted from the figure, but abstract measurement points are shown with circled 'M' symbol. The components are code sections used for configuration of the mechanism and deployment control structures. Regarding the identity data, information from the set of all identity information is needed for the uniqueness calculation. See the associated metric in [14]. If for example, we investigate the identity uniqueness of user name–password pair mechanism, the identity uniqueness CC metric is used to check if sufficient end-user password strength is deployed in the SuI. Password entropy metrics from the NIST (U.S. National Institute of Standards and Technology) Electronic Authentication Guideline [23] can be used. Moreover, the experimentation-based password creation policies suggested Weir et al. [24] can be applied in the metrics development. Later, the leaf nodes resulting from the decomposition process should include concrete measurable properties. Simple examples of BMs and DMs are listed in Table III.

TABLE III. EXAMPLES OF SECURITY CONFIGURATION BASE MEASURES AND DERIVED MEASURES [25]

Base Measure	**Derived Measure**
User authentication mode	Configuration command check: `auth_mechanisms = plain login cram-md5`
Denial of plaintext authentication without encryption	Configuration command check in Dovecot configuration file: `disable_plaintext_auth`
Mail backup up-to-datedness	Check appropriate use of the `rsync` application: `rsync –a /home/user/Maildir /media/backupdrive/mail`

In practice, the correct security configuration and deployment of CuIs is part of a wider goal, *correct system configuration*. Although not all system configuration issues are related directly to security, misconfigured systems are a significant source of vulnerabilities. For example, misconfiguration is related to a great extent to end-user visible downtime [26], thereby affecting to availability dimension of security. The suggested strategy for the decomposition of correct system configuration measurement objectives is similar to the approach of Table II. However, the scope of investigation is larger: instead of the CuI, the target is the whole SuI, which can be, e.g., the security-relevant part of the whole system, or the security-relevant part of it associated with the CuI. Because of the complexity and dynamicity of systems, the problem of measuring the correct system configuration can be a resource-demanding task. Self-managing configuration monitoring [27] can be utilized.

B. Direct Partial SE

Direct partial SE measurements can be carried out, e.g., by penetration testing or by monitoring during the system operation. Table IV shows a proposed strategy for direct partial SE metrics development.

Example (*Direct partial SE metrics vs. assurance metrics*) In [28], authentication metrics from [14] and (ii) the Authentication LoA (Level of Assurance) metrics for by the NIST [23] were compared. The metrics in [14] included *AIU*

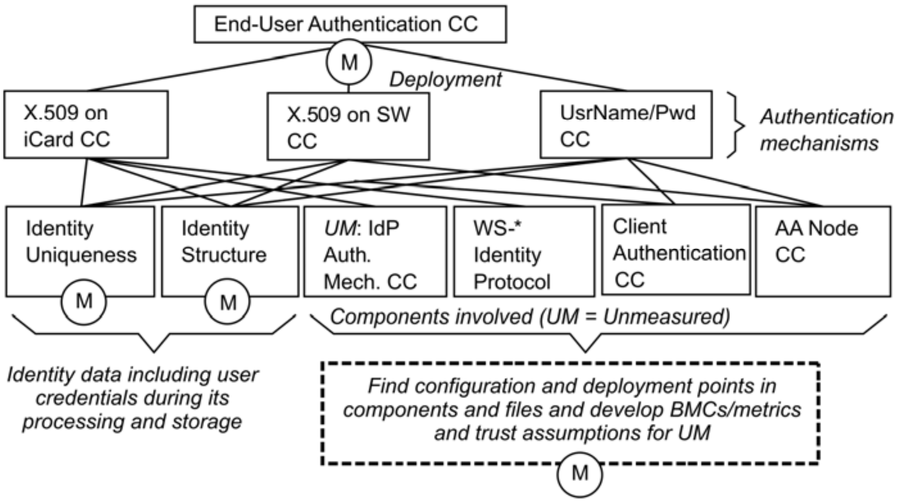
End-User Authentication CC
M
Deployment
X.509 on iCard CC
X.509 on SW CC
UsrName/Pwd CC
Authentication mechanisms
Identity Uniqueness
Identity Structure
UM: IdP Auth. Mech. CC
WS-* Identity Protocol
Client Authentication CC
AA Node CC
M
M
Components involved (UM = Unmeasured)
Identity data including user credentials during its processing and storage
Find configuration and deployment points in components and files and develop BMCs/metrics and trust assumptions for UM
M

Figure 5. A high-level plan of SCMO decomposition for end-user authentication CC.

(Authentication Identity Uniqueness), *AIS* (Authentication Identity Structure), *AII* (Authentication Identity Integrity), *AMR* (Authentication Mechanism Reliability), and *AMI* (Authentication Mechanism Integrity). These metrics correspond directly SEOs in authentication. However, this approach requires enough *operational-time* evidence to be attainable and available, making the measurement only partial. For example, *AIU* measurement requires knowledge about the total number of non-unique ID information values. This kind of information is difficult to obtain from an IdP, or even in some cases, is not known. Non-unique IDs might originate from faulty identity management procedures or algorithmic flaws. In the absence of direct security metrics, indirect metrics, or security assurance metrics, can be used to manage evidence from testing and monitoring activities. The LoA metrics are an example for this. As noted in [28], assurance metrics do not offer as strong SE evidence as direct SE metrics.

TABLE IV. STRATEGY 2: DIRECT PARTIAL SE

Stage	Additions to the general methodology
A1	SMOs are SEOs. Develop them as complete as possible. Even though only partial evidence is available, it is important to identify the overall objectives and gaps of evidence.
A2–A3	Develop an SEAM based on realistic conditions and aim at a sufficient measurement time. Penetration testing and incident statistics offer direct evidence. Incorporate qualitative interpretation mechanisms to the SEAM to cope with this information.
A4	Relevant components are the parts of the CuI, from which the testing and monitoring tools can be made to gather information, inside and outside the SuI.
A6	Investigate and select metrics which can be utilized for SE measurement. Measurement points for monitoring have to be available for longer period in time. in effectiveness measurements because security incident frequency can vary a lot. Note that measurement points for SE often contain data from outside the CuI or the SuI.
A7	Develop BMCs aiming at feasible metrics or use of available metrics. Detailed reference requirements can be used to define the effectiveness criteria.

C. *Software and System Quality Associated to the CuI*

In software-intensive systems, good SW&SQ (Software and System Quality) is a core enabling factor for SE. SW&SQ SMO decomposition *associated* to the CuI can be developed using the basic strategy of Table V. By 'association' we denote software and system quality of the actual CuI implementation and deployment, especially confidentiality, integrity and availability of the information and functionality protected by the CuI. The actual metrics in software and system quality may be based, e.g., on (i) source code analysis results, (ii) unit and system testing results, (iii) effort spent on engineering and testing the CuI, (iv) the verified lack of the vulnerabilities associated to the CuI according to information found from vulnerability databases, libraries, and earlier results. In the latter case, the vulnerability information is used as a reference model. It is important to investigate *overall* SW&SQ of the SuI, in addition to the SW&S quality associated to the CuI. It is very difficult to try to make a classification between what is associated to the CuI and what is not. In practice, this classification can be too resource-consuming and is not so relevant to practical security decision-making. The overall quality should be investigated as its own metrics hierarchy, separate from the security metrics. However, interdependencies should be identified between the overall quality and SE metrics hierarchies. Ouedraogo et al. [29] present a generic approach for security assurance metrics, with four main metrics dimensions: coverage, depth, rigor and the independence of verification. SW&SQ can be measured, e.g., using this Ouedraogo et al.'s taxonomy.

TABLE V. STRATEGY 3: SOFTWARE AND SYSTEM QUALITY ASSOCIATED WITH CUI

Stage	Explanation
A1	SMOs are based on good SW&SQ. However, their connection to SE should be analyzed, if possible.
A2–A3	Design or utilize a pre-existing SEAM emphasizing adequate confidentiality, integrity and availability of the CuI, and incorporate SW&SQ objectives into it.
A3	Applicable vulnerability databases offer important knowledge of SW&SQ. A reference model based on the robustness to prioritized vulnerabilities can be used.
A4	Identify components where evidence for quality can be gathered.
A6	Measurement points are typically focused on SW&SQ testing activities.

V. INTEGRATED STRATEGIES

Often Strategies 1–3 are not enough for practical decision-making. Compliance with best practices and regulations, and the tradeoff with security efficiency have to be considered too.

A. *'Pure' SE*

As discussed above, many factors contribute to the actual SE level of the SuI or the CuI. In software-intensive systems, SE measurements incorporate at least the factors discussed in Strategies 1–3. A strategy for the decomposition of 'pure' SE objectives is based on integration of the above strategies. If 'pure' SE is the SMO, compliance requirements or other security requirements are not necessarily used as reference model. However, software and system quality SMO decomposition can include relevant vulnerability information. In practice, 'pure' SE is not the only goal, due to the constrained resources and incompatibility of legal and other compliance requirements.

TABLE VI. STRATEGY 4: 'PURE' SE

Stage	Additions to the general methodology
A1	Carry out Stage A1 of Strategies 1–3 concentrating on SEOs.
A2–A3	Carry out Stages A2–A3 concentrating on prioritizing the resulting integrated SEAM from Strategies 1–3 . In general, the SEOs should be the highest priority. Consequently, the 'skeleton' of the SEAM should emphasize SE. The prioritization of the other SMOs is more case-dependent. Some objectives of Strategies 1–3 are overlapping, and some even conflicting. Identify the main overlaps and conflicts, and solve them *in favor of SE*.
A4–A7	Carry out Stages A4–A7 based on the prioritized integrated SEAM.

Example (*Application of 'pure' SE metrics*) 'Pure' SE metrics model can be used to increase the quality of RA. By identifying security controls which are the best to mitigate the risks, it is possible to set a reference level. Practical choices and constraints should be incorporated to the model to express the current security level. SE evidence should be gathered and fed back to the RA process, enabling evidence-based quality control of the RA.

B. SE vs. Compliance

Often, the aim of system developer is to ensure adequate SE, and best practices, likes standards and recommendations, are used to support this aim. Standards and recommendations very seldom are applicable directly to a particular SuI: they are meant for a wider use, and are therefore quite abstract. Examples of best practices include: the Common Criteria [9], SSE-CMM [30], ISO/IEC 27000 standard series [5], and COBIT [31]. It is important to note that although best practices give often valuable guidance in the design of security solutions, well-established risk-driven objectives are more valuable. If there are conflicts between SEOs and BPOs, the confidence on them should be compared and the solution offering higher confidence should win. In order to allow room for quality improvement of metrics, later adoption of risk-driven metrics should be supported, even though best practice-based metrics are used initially. Compliance with legal requirements and regulations (RMOs), and with other similar security requirements originating outside of the SuI developer organization are often important SMO categories. Table VII suggests an integrated strategy for compliance issues.

TABLE VII. STRATEGY 5: SE / COMPLIANCE TO BEST PRACTICES OR REGULATIONS

Stage	Additions to the general methodology
A1	Prioritize SEOs and BPOs. In case of unclear priority, well-established risk-driven SEOs should obtain priority over BPOs. If the SEOs are not yet well-established, BPOs can be followed, ensuring that it possible to later adopt risk-driven objectives. RMOs are typically of high priority. However, if evidence can be obtained that enforcing an SEO instead of an RMO results into better SE, the RMO should be questioned and reactions should be planned accordingly.
A2	Use SEOs to define the 'skeleton' of the SEAM. Incorporate BPOs and RMOs into it.
A3	Identify the potential differences between the SEOs, BPOs, RMOs, and the SEAMs. Decide how the differences should be treated and traced. If a BPO conflicts with a SEO, the withdrawal from the use of the BPO should be carefully considered. In a case of RMO–SEO conflict, potential further communication on the issue with the regulating party should be planned.
A4–A7	Assign SEOs, RMOs and BPOs in priority order as initial nodes for decomposition. However, use labeling to distinguish between SEOs, RMOs and BPOs in all nodes of the decomposition.

C. SE vs. Security Efficiency

In practice, software developers are interested in implementing sufficiently effective security solutions in an efficient way. Therefore, it is useful to have a systematic strategy for developing metrics for both objectives, enabling *informed tradeoff decision-making,* see Table VIII.

Example (*SE / Security Effectiveness Tradeoff*) A large number of sensors are often used in IoT (Internet of Things) applications to gather information. The applications include home automation, smart grid control, and car automation. Many sensors afford to consume much power. Therefore, complex communication and calculation in the sensors is not possible in many cases. The constraints can often result into poor identity strength from authentication perspective. The SE / Security effectiveness tradeoff should be modeled and visualized by parallel branches of both tradeoff objectives, enabling comparison of different scenarios. Tradeoff decision vary depending on the power consumption limits of each application. The resulting SE should be systematically visualized in the SE branch to clearly show the risk accepted, in connection of the achieved power (and security) efficiency.

TABLE VIII. STRATEGY 6: SE / SECURITY EFFICIENCY TRADEOFF

Stage	Additions to the general methodology
A1	Prioritize SEOs and SEyOs *separately*.
A2	Build the 'skeleton' of the SEAM based on SEOs. Incorporate SEyOs to the relevant interaction points of the SEAM. This process should be continued during the actual decomposition stages (A4–A7).
A3	Identify potential differences between SEOs and SEyOs, and decide how the differences should be treated and traced. For example, plan additional metrics to show possible redundancy / new contribution in the differences.
A4–A7	During the decomposition process, build a metrics hierarchy, which is as complete as possible from SEO perspective, with SEyOs incorporated at relevant nodes. It should be possible to compare the implications of chosen security efficiency levels on SE.

VI. RELATED WORK

Haddad et al. [32] introduced an abstract security model called AP (Assurance Profile), a template to define a common set of assurance measurement needs of an SuI. According to [32], APs contain the following information: (i) description of the target system, (ii) security problem definition, (iii) compliance claims, (iv) SMOs, (v) security requirements, (vi) assurance measurement objectives, (vii) assurance measurement requirements, and (viii) Security Assurance Views (SAVs). Each SAV is a representation of assurance information related to a specific goal, such as regulation compliance. Basili et al. [33] defined GQM (Goal Question Metrics), a three-level decomposition approach for refining specification of software measurements. The highest level is conceptual level (goals), the next one operational (question), and the third one quantitative level (metric). The AP and the GQM definitions lack strategies or heuristics to define their security-relevant content. The strategies introduced in the present study can be integrated to both of them. Kirkman [34] discusses the potential challenges of requirement decomposition in general: excessive 'subsystem' decomposition, insufficient decomposition, unsourced requirements, excessive hierarchy, insufficient hierarchy and change management. Koopman [35] introduces taxonomy for decomposition strategies for design, focusing on structures, behaviors, and goals. He discusses decomposition of goals and notes that "Excessive focus on goals can promote 'gaming' on the part of designers. Loopholes in the goal statements may be exploited, and it is possible to create designs that meet all stated goals, but fail on the implicit goals of 'does it work?'". This is an important consideration especially in decision-making involving tradeoff balance between SEOs and SEyOs.

VII. Conclusions and Future Work

We have proposed six strategies for security measurement objective decomposition for top-down risk-driven security metrics development and management. The basic strategies address security configuration correctness, direct partial security effectiveness, and software and system quality. The integrated strategies were formed to support decision-making aiming at compliance with best practices and regulations, pure security effectiveness, and security effectiveness–efficiency tradeoff. Metrics for software and system quality are needed because the granularity of security control concept is not detailed enough. In our approach, an abstract security effectiveness model is used to guide the decomposition process. This process itself is iterative, where the aim is to find concrete measurable properties, which relate to the original measurement objectives. Relevant system components to be used in the measurement are associated with relevant nodes of the decomposition. Enough emphasis on security effectiveness is important in all strategies, even though other measurement objectives should be taken appropriately into account. Our future work includes a formal definition of decomposition strategies, and gathering practical experience on the application of them in industrial pilot studies.

References

[1] D. S. Herrmann, "Complete Guide to Security and Privacy Metrics – Measuring Regulatory Compliance, Operational Resilience and ROI," Auerbach Publications, 2007, 824 p.

[2] A. Jaquith, "Security Metrics: Replacing Fear, Uncertainty and Doubt," Addison-Wesley, 2007.

[3] N. Bartol, B. Bates, K.M. Goertzel, and T. Winograd, "Measuring Cyber Security and Information Assurance: A State-of-the-art Report," Information Assurance Technology Analysis Center IATAC, May 2009.

[4] V. Verendel, "Quantified Security is a Weak Hypothesis: A Critical Survey of Results and Assumptions," New Security Paradigms Workshop, Oxford, U.K., 2009, pp. 37–50.

[5] ISO/IEC 27000:2009. Information Technology – Security Techniques – Information Security Management Systems – Overview and Vocabulary. International Organization for Standardization and the International Electrotechnical Commission, 2009.

[6] R. Savola, "Security Metrics Taxonomization Model for Software-Intensive Systems," Journal of Information Processing Systems, Vol. 5, No. 4, Dec. 2009, pp. 197–206.

[7] W. Jansen, "Directions in Security Metrics Research," U.S. National Institute of Standards and Technology, NISTIR 7564, Apr. 2009, 21 p.

[8] ITSEC. Information Technology Security Evaluation Criteria (ITSEC), Version 1.2, Commission for the European Communities, 1991.

[9] ISO/IEC 15408-1:2005. Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and General Model, International Organization for Standardization and the International Electrotechnical Commission, 2005.

[10] L. Hayden, "IT Security Metrics – A Practical Framework for Measuring Security & Protecting Data," McGraw-Hill, 2010.

[11] G. McGraw, "Software Security – Building Security In," Addison-Wesley, 2009.

[12] M. Howard and S. Lipner, "The Security Development Lifecycle," Microsoft Press, 2006.

[13] IEC 62628 Ed. 1.0. "Guidance on Software Aspects of Dependability." International Electrotechnical Commission (Committee Draft), 2010.

[14] R. Savola and H. Abie, "Development of Measurable Security for a Distributed Messaging System," International Journal on Advances in Security, Vol. 2, No. 4, 2009, pp. 358–380 (Published March 2010).

[15] R. Savola and H. Abie, "Identification of Basic Measurable Security Components for a Distributed Messaging System," Proc. 3rd Int. Conf. on Emerging Security Information, Systems and Technologies (SECURWARE '09), pp. 121–128.

[16] H. Abie, I. Dattani, M. Novkovic, J. Bigham, S. Topham, and R. Savola, "GEMOM – Signficant and Measurable Progress Beyond the State of the Art," Proc. 3th Int. Conf. on Systems and Networks Communications (ICSNC '08), pp. 191–196.

[17] M. Ouedraogo, D. Khadraoui, B. de Rémont, E. Dubois, H. Mouratidis, "Deployment of a Security Assurance Monitoring Framework for Telecommunication Service Infrastructure on a VoIP system," NTMS '08, 5–7 Nov. 2008.

[18] R. Savola, H. Pentikäinen, and M. Ouedraogo, "Towards Security Effectiveness Measurement utilizing Risk-Based Security Assurance," Proc. ISSA 2010.

[19] R. Savola and P. Heinonen, "A Visualization and Modeling Tool for Security Metrics and Measurements Management," Proc. ISSA 2011.

[20] OASIS Standard 200401. "Web Services Security: SOAP Message Security 1.0," Organization for the Advancement of Structured Information Standards, 2004.

[21] ITU Recommendation X.509 (08/05), "Information Technology – Open Systems Interconnection – The Directory: Public-key and Attribute Certificate Frameworks," International Telecommunication Union, 2005.

[22] L. Blasi, R. Savola, H. Abie, and D. Rotondi, "Applicability of Security Metrics for Adaptive Security Management in a Universal Banking Hub System," Proc. 4th European Conference on Software Architecture: Companion Volume, pp. 197–204.

[23] W.E. Burr et al., "Electronic Authentication Guideline," U.S. National Institute of Standards and Technology Special Publication 800-63-1, December 2011, 121 p.

[24] M. Weir, S. Aggarwal, M. Collins, H. Stern, "Testing Metrics for Password Creation by Attacking Large Sets of Revealed Passwords," CCS'10, pp. 162–175.

[25] R. Savola, T. Kanstrén, H. Pentikäinen, P. Jurmu, M. Myllyaho, K. Hätönen, "Utilizing a Risk-Driven Operational Security Assurance Methodology and Measurement Architecture – Experiences from a Case Study," Proc. 8th Int. Conf. on Networking and Services (ICNS '12), 2012, pp. 134-142.

[26] D. Oppenheimer, A. Ganapathi, and D. Patterson, "Why do Internet Services Fail, and What can be done about it?" Proc. 4th USENIX Symposium on Internet Technologies and Systems (USITS '03), 2003.

[27] E. Kıcıman and Y.-M. Wang, "Discovering Correctness Constraints for Self-Management of System Configuration," Proc. 1st Int. Conf. Autonomic Computing (ICAC '04), pp. 28–35.

[28] L. Blasi, R. Savola, H. Abie, and D. Rotondi, "Applicability of Security Metrics for Adaptive Security Management in an Univeral Banking Hub System," Proc. 4th European Conf. on Software Architecture: Companion Volume, 2010, pp. 1974–204.

[29] M. Ouedraogo, R. Savola, H. Mouratidis, D. Preston, D. Khadraoui, and E. Dubois, "Taxonomy of quality metrics for assessing assurance of security correctness," In: Software Quality Journal, Online First, 30 Nov. 2011, 30 p.

[30] ISO/IEC 21827:2003, "Information technology – Systems Security Engineering – Capability Maturity Model (SSE-CMM)," ISO/IEC 2003.

[31] IT Governance Institute, "CobiT 4.1," 2007, 213 p.

[32] S. Haddad, S. Dubus, A. Hecker, T. Kanstrén, B. Marquet and R. Savola, "Operational Security Assurance Evaluation in Open Infrastructures," Proc. CRiSIS 2010, pp. 100–105.

[33] V.R. Basili, G. Caldiera, H.D. Rombach, "Goal Question Metric Paradigm," Encyclopedia of Software Engineering, Wiley, 1994.

[34] D. Kirkman, "Requirement Decomposition and Traceability," Requirements Engineering, Vol. 3, No. 2, 1998, pp. 107–114.

[35] P. Koopman, "A Taxonomy of Decomposition Strategies based on Structures, Behaviors, and Goals," Design Theory & Methodology '95, 1995.